TP
Test Plan
Sapori di Unisa

Riferimento	NC1_TP
Versione	1.0
Data	15/12/2023
Destinatario	Prof. Carmine Gravino
Presentato da	Antonio Facchiano, Gianmarco Riviello, Salvatore Ruocco, Simone Vittoria
Approvato da	

Revision History

Data	Versione	Descrizione	Autori
05/12/2023	0.1	Introduzione, Relazione con altri documenti, Panoramica del sistema, Materiale di testing, Approccio	Salvatore Ruocco
05/12/2023	0.3	Funzionalità da testare e non, Pass/Fail Criteria	Gianmarco Riviello
09/12/2023	0.5	Test Frame	Gianmarco Riviello Salvatore Ruocco
15/12/2023	1.0	Revisione e delibera documento	Gianmarco Riviello Salvatore Ruocco Simone Vittoria Antonio Facchiano

Sommario

1 Introduzione

1.1 Obiettivi

Sapori di Unisa è un supermercato che ha deciso di rinnovarsi digitalizzando la gestione della propria attività. I dirigenti hanno rilevato la necessità di modernizzare i processi; pertanto, hanno optato per l'introduzione di sistemi informatici per gestire le finanze e il magazzino. In particolare, hanno individuato delle discrepanze nella gestione dovute a errori umani, e per questo motivo hanno scelto di minimizzare tali errori adottando un software appositamente progettato.

Il sistema che si vuole realizzare ha come obiettivo principale quello di ottimizzare la gestione del magazzino e di avere un maggior controllo riguardo gli aspetti finanziari dell'attività. Attraverso la piattaforma, i dipendenti del supermercato potranno visionare in tempo reale la disponibilità dei prodotti presenti in magazzino e di conseguenza intraprendere le giuste azioni. Inoltre, il sistema consente di tener traccia degli aspetti finanziari e in particolare di controllare gli utili e le spese del supermercato.

Volendo rispettare questi obiettivi, non è possibile trascurare l'importanza di fornire al cliente un prodotto quanto più affidabile possibile. Sarà obiettivo di questo documento fornire strategie necessarie al team per verificare il comportamento di quello che sarà il sistema implementato rispetto a quanto descritto nei documenti di analisi dei requisiti e system design.

Detto ciò, questo documento fa da apripista per l'intero team nelle attività di testing dove sarà importante individuare failure e correggere appena possibile i relativi fault.

1.2 Definizioni, acronimi e abbreviazioni

Vengono riportati di seguito alcune definizioni presenti nel documento corrente:

- **SDD:** System Design Document
- **RAD:** Requirements Analysis Document
- **TP**: Test Plan

1.3 Riferimenti

- Libro di testo utilizzato durante il corso: **Object-Oriented Software Engineering Using UML, Patterns, and Java**
- **Statement Of Work**
- **RAD**
- **SDD**

2 Relazione con gli altri documenti

Documenti correlati al TP:

- **RAD**
  La pianificazione dei casi di test non può non dipendere da quanto è stato descritto in fase di raccolta e analisi dei requisiti; quindi, si farà riferimento alle specifiche descritte nel RAD, in particolar modo ai requisiti funzionali, agli scenari e ai casi d'uso.
- **SDD**
  Qui la nostra attenzione sarà rivolta verso la tabella che illustra i servizi dei sottosistemi per definire quali funzionalità saranno argomento di testing.

3 Panoramica del sistema

L'architettura del sistema è **three-tier**. Uno dei motivi che hanno influenzato tale scelta è sicuramente il background degli sviluppatori, i quali hanno già avuto a che fare con tale architettura. Inoltre, essa è molto indicata per lo sviluppo di applicazioni web, poiché si ha una netta separazione tra interfacce utente, logica applicativa e persistenza dei dati. Queste caratteristiche garantiscono una serie di qualità essenziali per lo sviluppo software odierno, su tutte la manutenzione e il riuso.

Per lo sviluppo del sistema verranno utilizzate le seguenti tecnologie.

Back-end	Front-end
• Java 21 • MySQL 5.6 • Apache Tomcat 10.1	• HTML5 • JSP 3.1 • JSTL 1.2.7 • CSS3 • JavaScript • JQuery 3.7.1

Per la gestione delle librerie verrà usato **Maven**.

4 Funzionalità da testare e non

Qui si elencheranno le features alle quali verrà applicato il testing per i vari sottosistemi.

- Gestione Magazzino:
  - Registra fornitura
- Gestione Vendite:
  - Vendita prodotti
- Autenticazione:
  - Login
  - Modifica pin
- Gestione Scaffali:
  - Aggiungi prodotto scaffale
- Gestione Finanze:
  - Visualizza bilancio
  - Imposta sconto

Le funzionalità di cui non si andrà ad effettuare attività di testing riguardano requisiti funzionali di bassa priorità; sono inoltre escluse le funzionalità che non prevedono input manuale da parte dell'utente - ad esempio attività riguardanti esclusivamente la visualizzazione di dati o semplici click di un bottone.

5 Pass/Fail criteria

Il testing viene eseguito per andare a stanare le varie anomalie del sistema proposto, in gergo tecnico **faults**, per poi andare a correggere con specifiche modalità d'approccio e strumenti le anomalie.

Il risultato atteso dei vari test case è confrontato con un oracolo (specifica del risultato atteso secondo i criteri progettuali, vincoli e requisiti funzionali richiesti).

In caso di successo nella correlazione dei dati previsti tra oracolo e test case effettivo si ha il successo del test, ***Pass.***

Nel caso contrario si ha un ***Fail***.

Per cercare di ottenere un software con 0 difetti si sa che c'è bisogno di una continua e precisa attività di verifica e convalida che va effettuata e applicata durante tutta la progettazione; per questo, il test ha dei requisiti ben precisi per essere ritenuto valido:

- Testare tutti i requisiti funzionali elencati nel punto 4;
- Eseguire test di regressione nel momento in cui si vanno ad aggiungere altre caratteristiche al sistema oppure modificando quelle già comprese in precedenza.

6 Approccio

Il testing dell'intero sistema ha tre fasi:

- testing di sistema
- testing di integrazione
- testing di unità

Prima della fase di implementazione del sistema, avverrà la progettazione dei casi di test di sistema, perfezionati in seguito nella loro fase di esecuzione.

Solo durante la fase di implementazione del sistema avverrà la progettazione dei casi di test di unità e di integrazione.

6.1 Testing di sistema

Questa fase ha come obiettivo la definizione di casi di test che determinino l'integrità del comportamento del sistema implementato con ciò che l'utente si aspetta; quindi, non è di primaria importanza avere a disposizione il codice sorgente. Per questa fase, infatti, si prevede l'utilizzo di una metodologia black-box. Nello specifico, si è deciso di adottare per il testing di sistema la metodologia Category Partition.

6.2 Testing di integrazione

Da effettuare prossimamente

6.3 Testing di unità

Da effettuare prossimamente

7 Materiale di testing

Come già anticipato, le attività di testing richiederanno il supporto dei documenti di progetto RAD, SDD in modo da individuare le componenti da testare.

L'esecuzione dei test dovrà avvenire su un server in locale, nel quale verrà eseguita l'applicazione. Su codesto server dovranno essere configurate le tecnologie usate per lo sviluppo del sistema.

Nella fase di perfezionamento del testing di sistema, verrà usato il tool Selenium, il quale automatizzerà le interazioni utente con le interfacce del sistema.

Mentre, Il testing di unità dovrà essere svolto tramite il tool JUnit, per il testing di integrazione verrà usato il framework Mockito.

Per il monitoraggio dei Branch Coverage è stato scelto il tool JaCoCo.

8 Test Cases

Per sviluppare i test cases sarà utilizzato il metodo del Category Partition. Questo metodo consiste nell'identificare, per ogni funzionalità da testare, dei parametri; per ogni parametro verranno individuate delle categorie, le quali poi saranno suddivise in scelte. Alle scelte verrà assegnato un valore.

Quindi, per ogni requisito funzionale, dobbiamo:

1. Individuare una serie di parametri (input)
2. Per ogni parametro, individuare le categorie
3. Per ogni categoria, individuare le scelte
4. Andare a costruire i Test Frame: combinazioni di scelte (sensate) da ogni categoria individuata
5. Per ogni Test Frame, indicare l'esito: *errato* se non consente il raggiungimento dell'obiettivo del requisito, altrimenti *corretto*
6. Per ogni Test Frame, si crea un Test Case, che assegna dei valori precisi alle scelte

I test cases verranno definiti nel documento di Test Cases Specification (TCS).

9 Specifiche dei test cases

9.1 Gestione Magazzino

TC_1.1 Registra fornitura

Parametro: Nome prodotto **Formato:** ^[a-zA-Z0-9\s]{2,255}$	
Categorie	**Scelte**
Lunghezza [LN]	**1:** Lunghezza <= 1 [errore] **2:** Lunghezza >= 2 && Lunghezza <= 255 [property **LunghezzaLN_OK**] **3:** Lunghezza >= 256 [errore]
Formato [FN]	**1:** rispetta il formato [property **FormatoFN_OK**] **2:** non rispetta il formato [errore]

Parametro: Marchio **Formato:** ^[a-zA-Z0-9\s]{2,255}$	
Categorie	**Scelte**
Lunghezza [LM]	**1:** Lunghezza <= 1 [errore] **2:** Lunghezza >= 2 && Lunghezza <= 255 [property **LunghezzaLM_OK**] **3:** Lunghezza >= 256 [errore]
Formato [FM]	**1:** rispetta il formato [property **FormatoFM_OK**] **2:** non rispetta il formato [errore]

Parametro: Prezzo di acquisto Formato: ^[0-9]{1,5}.[0-9]{2}$	
Categorie	**Scelte**
Valore [VA]	**1:** Valore <= 0.00 [errore] **2:** Valore >= 0.01 && Valore <= 99999.99 [property **ValoreVA_OK**] **3:** Valore >= 100000.00 [errore]
Formato [FA]	**1:** rispetta il formato [property **FormatoFA_OK**] **2:** non rispetta il formato [errore]

Parametro: Quantità **Formato**: ^\d+$	
Categorie	**Scelte**
Formato [FQ]	**1:** rispetta il formato [property **FormatoFQ_OK**] **2:** non rispetta il formato [errore]
Valore [VQ]	**1:** Valore <= 0 [errore] **2:** Valore >= 1000000 [errore] **3:** Valore >= 1 && Valore <= 999999 [property **Valore_VQ_OK**]

Parametro: Data di scadenza	
Categorie	**Scelte**
Valida [VD]	**1:** valida [property **ValidaVD_OK**] **2:** non valida [errore]
Correttezza [CD]	**1:** dataDiScadenza < dataCorrente[errore] **2:** dataDiScadenza >= dataCorrente[property **CorrettezzaCD_OK**]

Parametro: Foto	
Categorie	**Scelte**
Estensione [EI]	**1:** estensione = PNG [property **EstensioneEI_OK**] **2:** estensione != PNG [errore]
Dimensioni [DI]	**1:** dimensioni >= 2 MB [errore] **2:** dimensioni < 2 MB [property **DimensioniDI_OK**]

Codice	Combinazione	Esito
TC_1.1.1	LN1	Errore: nome prodotto troppo corto
TC_1.1.2	LN3	Errore: nome prodotto troppo lungo
TC_1.1.3	LN2.FN2	Errore: nome prodotto formato non rispettato
TC_1.1.4	LN2.FN1.LM1	Errore: marchio troppo corto

TC_1.1.5	LN2.FN1.LM3	Errore: marchio troppo lungo
TC_1.1.6	LN2.FN1.LM2.FM2	Errore: marchio non rispettato
TC_1.1.7	LN2.FN1.LM2.FM1.VA1	Errore: prezzo acquisto insufficiente
TC_1.1.8	LN2.FN1.LM2.FM1.VA3	Errore: prezzo acquisto eccedente
TC_1.1.9	LN2.FN1.LM2.FM1.VA2.FA2	Errore: prezzo acquisto formato non valido
TC_1.1.10	LN2.FN1.LM2.FM1.VA1.FA1.FQ2	Errore: quantità non è un numero
TC_1.1.11	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ1	Errore: quantità minore di 0
TC_1.1.12	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ2	Errore: quantità eccedente
TC_1.1.13	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ3.VD2	Errore data scadenza formato non valido
TC_1.1.14	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ3.VD1.CD1	Errore: data scadenza precedente a quella attuale
TC_1.1.15	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ3.VD1.CD2.EI2	Errore: estensione immagine diversa da PNG
TC_1.1.16	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ3.VD1.CD2.EI1.DI1	Errore: dimensioni immagine superiore a 2 MB
TC_1.1.17	LN2.FN1.LM2.FM1.VA1.FA1.FQ1.VQ3.VD1.CD2.EI2.DI2	Corretto

9.2 Gestione Vendite

TC_2.1 Vendita Prodotti

Parametro: ID Prodotto	
Categorie	**Scelte**
Valido [VDID]	**1:** valida [property **ValidaVDID_OK**] **2:** non valida [errore]

Parametro: Quantità **Formato**: ^\d+$	
Categorie	**Scelte**
Formato [FQ]	**1:** rispetta il formato [property **FormatoFQ_OK**] **2:** non rispetta il formato [errore]
Valore [VQ]	**1:** Valore <= 0 [errore] **2:** Valore >= 1000000 [errore] **3:** Valore >= 1 && Valore <= 999999 [property **Valore_VQ_OK**]
QuantitàScaffali [QS]	**1:** quantitàScaffali – ValoreVQ >= 0 [property **QuantitàScaffaliQS_OK**] **2:** quantitàScaffali – ValoreVQ < 0 [errore]

Codice	Combinazione	Esito
TC_2.1.1	VDID2	Errore nella selezione del prodotto.
TC_2.1.2	VDID1.FQ2	Errore: formato inserito errato.
TC_2.1.3	VDID1.FQ1.VQ1	Errore: la quantità inserita è sotto il limite minimo consentito.
TC_2.1.4	VDID1.FQ1.VQ2	Errore: la quantità inserita è sopra il massimo limite consentito.
TC_2.1.5	VDID1.FQ1.VQ3.QS2	Errore: la quantità selezionata del prodotto non è disponibile.
TC_2.1.6	VDID1.FQ1.VQ3.QS1	Corretto.

9.3 Autenticazione

TC_3.1 Login

Parametro: Pin **Formato**: ^[0-9]{4}$	
Categorie	**Scelte**
Formato [FQ]	**1:** rispetta il formato [property **FormatoFQ_OK**] **2:** non rispetta il formato [errore]
Match [MC]	**1:** Match con pin = false [errore] **2:** Match con pin = true [property **MatchMQ_OK**]

Codice	Combinazione	Esito
TC_3.1.1	FQ2	Errore: Formato pin non rispettato
TC_3.1.2	FQ1.MC1	Errore: Nessun account corrisponde a questo pin
TC_3.1.3	FQ1.MC2	Corretto

TC_3.2 Modifica Pin

Parametro: Pin **Formato**: ^[0-9]{4}$	
Categorie	**Scelte**
Formato [FQ]	**1:** rispetta il formato [property **FormatoFQ_OK**] **2:** non rispetta il formato [errore]
Match [MC]	**1:** Match con pin attivo in quel momento = true[errore] **2:** Match con pin != true [property **MatchMC_OK**]

Codice	Combinazione	Esito
TC_3.2.1	FQ2	Errore: Formato pin non rispettato
TC_3.2.2	FQ1.MC1	Errore: Quest'account ha già impostato questo pin, deve essere diverso.
TC_3.2.3	FQ1.MC2	Corretto

9.4 Gestione Scaffale

TC_4.1 Aggiungi scaffale

Parametro: Quantità **Formato**: ^\d+$	
Categorie	**Scelte**
Valore [VQ]	**1:** Valore <= 0 [errore] **2:** Valore >= 1000000 [errore] **3:** Valore >= 1 && Valore <= 999999 [property **ValoreVQ_OK**]
Formato [FQ]	**1:** rispetta il formato [property **FormatoFQ_OK**] **2:** non rispetta il formato [errore]
QuantitàMagazzino [QM]	**1:** quantità magazzino – ValoreVQ >= 0 [property **QuantitàMagazzinoQM_OK**] **2:** quantità magazzino – ValoreVQ < 0 [errore]

Codice	Combinazione	Esito
TC_4.1.1	VQ1	Errore: valore quantità troppo basso
TC_4.1.2	VQ2	Errore: valore quantità troppo alto
TC_4.1.3	VQ3.FQ2	Errore: formato quantità non valido
TC_4.1.4	VQ3.FQ1.QM2	Errore: quantità non presente in magazzino
TC_4.1.5	VQ3.FQ1.QM1	Corretto

9.5 Gestione Finanze

TC_5.1 Visualizza Bilancio

Parametro: Data inizio	
Categorie	**Scelte**
Valida [VI]	**1:** valida [property **ValidaVI_OK**] **2:** non valida [errore]

Parametro: Data fine	
Categorie	**Scelte**
Valida [VF]	**1:** valida [property **ValidaVF_OK**] **2:** non valida [errore]
Correttezza [CF]	**1:** dataFine < dataInizio [errore] **2:** dataFine >= dataInizio [property **CorrettezzaCF_OK**]

Codice	Combinazione	Esito
TC_5.1.1	VI2	Errore: data inizio non valida
TC_5.1.2	VI1.VF2	Errore: data fine non valida
TC_5.1.3	VI1.VF1.CF1	Errore: data fine precede quella di inizio
TC_5.1.4	VI1.VF1.CF2	Corretto

TC_5.2 Imposta sconto

Parametro: ID prodotto	
Categorie	**Scelte**
Valido [VP]	**1:** valido [property **ValidoVP_OK**] **2:** non valido [errore]
GiàScontato [GSP]	**1:** prodotto non in sconto [property **GiàScontatoGSP_OK**] **2:** prodotto già in sconto [errore]

Parametro: Data inizio	
Categorie	**Scelte**
Valida [VI]	**1:** valida [property **ValidaVI_OK**] **2:** non valida [errore]
Futuro [FI]	**1:** dataInizio > dataCorrente [property **CorrettezzaFI_OK**] **2:** dataInizio <= dataCorrente[errore]

Parametro: Data fine	
Categorie	**Scelte**
Valida [VF]	**1:** valida [property **ValidaVF_OK**] **2:** non valida [errore]
Correttezza [CF]	**1:** dataFine > dataInizio [property **CorrettezzaCF_OK**] **2:** dataFine <= dataInizio [errore]

Parametro: Prezzo Scontato Formato: ^[0-9]{1,5}.[0-9]{2}$	
Categorie	**Scelte**
Valore [VA]	**1:** Valore <= 0.00 [errore] **2:** Valore >= 0.01 && Valore < Prezzo standard [property **ValoreVA_OK**] **3:** Valore >= Prezzo Standard [errore]
Formato [FP]	**1:** rispetta il formato [property **FormatoFP_OK**] **2:** non rispetta il formato [errore]

Codice	Combinazione	Esito
TC_5.2.1	VP2	Errore: id prodotto non valido
TC_5.2.2	VP1.GSP2	Errore: prodotto già in sconto
TC_5.2.3	VP1.GSP1.VI2	Errore: data inizio non valida
TC_5.2.4	VP1.GSP1.VI1.FI2	Errore: data inizio antecedente a quella odierna
TC_5.2.5	VP1.GSP1.VI1.FI1.VF2	Errore: data fine non valida
TC_5.2.6	VP1.GSP1.VI1.FI1.VF1.CF2	Errore: data fine antecedente a quella di inizio
TC_5.2.7	VP1.GSP1.VI1.FI1.VF1.CF1.VA1	Errore: valore del prezzo scontato troppo basso.
TC_5.2.8	VP1.GSP1.VI1.FI1.VF1.CF1.VA3	Errore: valore del prezzo scontato più alto di quello standard.
TC_5.2.9	VP1.GSP1.VI1.FI1.VF1.CF1.VA2.FP2	Errore: formato del prezzo scontato non rispettato
TC_5.2.10	VP1.GSP1.VI1.FI1.VF1.CF1.VA2.FP1	Corretto